
JETSTACK
Presented by:
Josh van Leeuwen
External CAs with Istio
5 August 2021
jetstack.io

Istio / External CA: Me

Josh Van Leeuwen
Senior Software Engineer

Istio / External CA: Contents

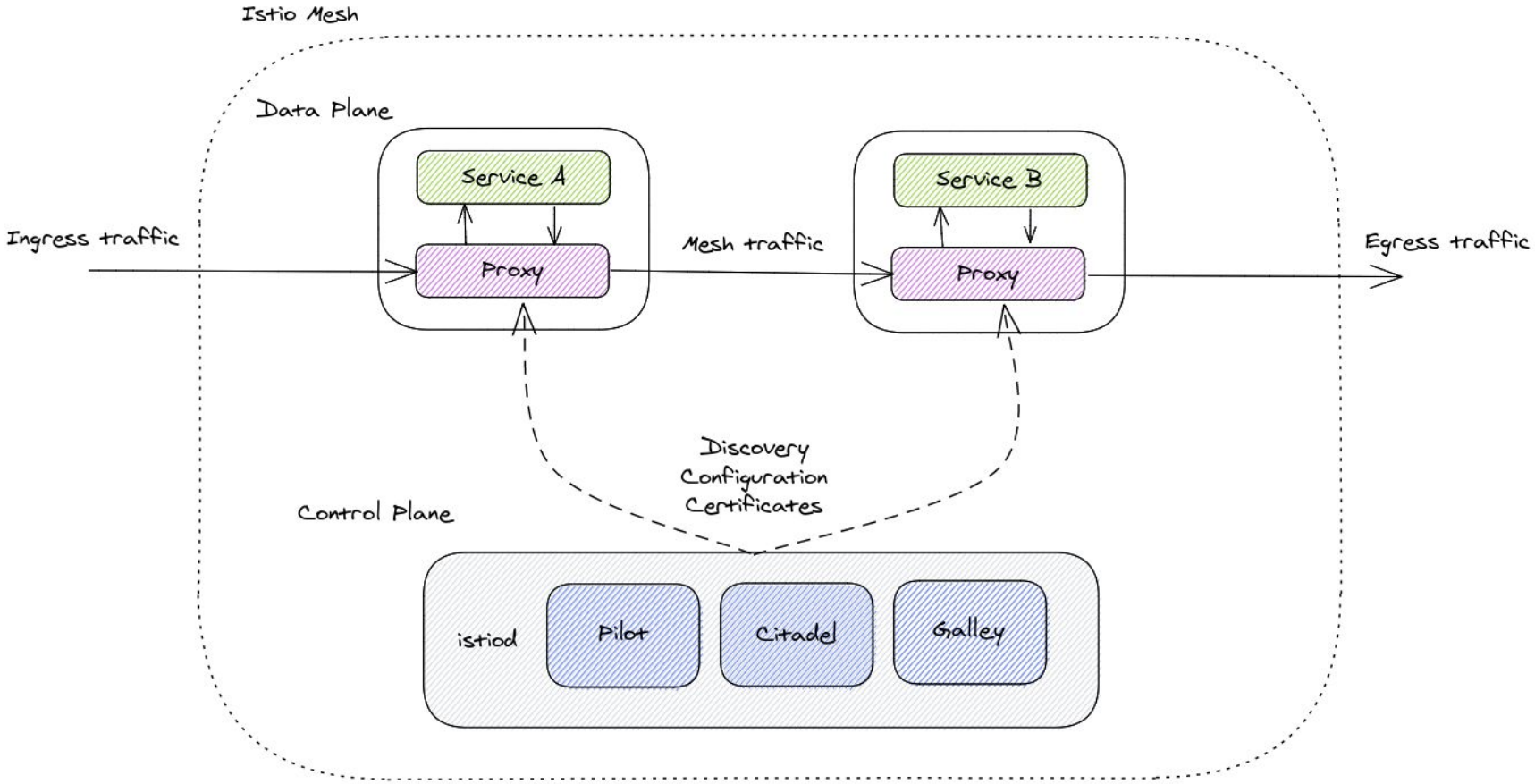
- Service Identity
- Trust Distribution
- Default installation
- External CAs

Istio / External CA: Contents

- Service Identity
- Trust Distribution
- Default installation
- External CAs
- Plugging in an External CA
- Using Kubernetes CSR for signing certificates
- Using cert-manager/istio-csr for signing certificates

Service Identity

Istio / External CA: Service Identity



Istio / External CA: Service Identity

- Istio uses X.509 certificates as identity documents
- Enables Mutual TLS (mTLS)

```
-----BEGIN CERTIFICATE-----
MIIDTDCCAjSgAwIBAgIQUDRiWJ9L2NGL3Ibxa6dk/TANBgkqhkiG9w0BAQsFADBA
MSswEwYDVQQKEwxjZXJ0LW1hbmFnZXIwFAYDVQQKEw1jbHVzdGVyLmxvY2FsMREw
DwYDVQQDEwhpc3Rpby1jYTAeFw0yMTA4MDMxMDM2MjBaFw0yMTExMDExMDM2MjBa
MEAxKzATBgNVBAoTDGNlcnQtbWFuYWdlcjAUBgNVBAoTDWNsdXN0ZXIubG9jYWwx
ETAPBgNVBAMTCGlzdGlvLWNhMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC
AQEAuM5xUppoXwoeWWGFhiviwkY5rDxriW5bt9ZO4qYf7/NiBjC68szMxls+XTn3
SPhx7u0FcBaI7C0a6rYNJxflNuUQ5E+BBjuUd0tnxU6UGIIERPb5ZDtsxZQJg9qh
eYkRgiDEAi5MKp0+O0uAh9JCFOSq3aZMVS9lQ5ib/S4gNzIiC5skpSVIBrXhtEZV
foi5VmcTOfS/kDR0avh+yJtxh5vQtUTLf7MpuUBNrhylHF+Q6+M/pD+SpoQwe221
KHfDLLxAeslHomJBtd2X40lx7JEv9CydjNBxw3eYN1SZ+SMxiJTBT6/KpzSbDOSY
d7MeH8q3S0kZj09bzhRjlEokBwIDAQABo0IwQDAOBgNVHQ8BAf8EBAMCAqQwDwYD
VR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUAmTumk1Xkr0eszilLALtnAcGGnMwDQYJ
KoZIhvcNAQELBQADggEBAC7BS7T9tx337XB0dNsUSL67apumNrfapEzHjxXxPg5F
dO5JFv7SqJFAh0m90yjaGL3uUKoEb+UCD+7u3J4QMbwCcIL5+i1WmFgg5Zan6FN5
7W7ImlHqH43Rxw/q6qBrAptQfseLkFQDbRhz6lSqKT7cMUATnik546Fpt2FSle0/
7+hYZafv+GMeUOageyZAfNVMWBvHH21G1+GNtQWdPf8od3L8QMHTAUoV5xyChR5L
1GjLfkh4gnhyZI1kR7kVMCTQZ2GR+RLwv9X+Rjai44Sidk7GTeRicwgu9eKKX1oO
XhihGpVDl38hJCHm2LBwEfrpH6s2Qk1ps4mqlZdmHeU=
-----END CERTIFICATE-----
```

```
Validity
    Not Before: Aug  3 11:06:31 2021 GMT
    Not After : Aug  3 11:11:31 2021 GMT
```

```
        X509v3 extensions:
            X509v3 Extended Key Usage:
                TLS Web Client Authentication, TLS Web Server Authentication
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Authority Key Identifier:
                keyid:02:64:EE:9A:4D:57:92:BD:1E:B3:38:A5:2C:02:ED:9C:07:06:1A:73

            X509v3 Subject Alternative Name:
                URI:spiffe://foo.bar/ns/istio-system/sa/istio-ingressgateway-service-account
```

Istio / External CA: Service Identity

Secure Production Identity Framework for Everyone
A SPIFFE ID is used to uniquely and specifically identify a service

spiffe://trust domain/workload identifier

spiffe://cluster.local/ns/ns-a/sa/app-a

Full specification at spiffe.io

SVID (SPIFFE Verifiable Document)

X.509 SVID

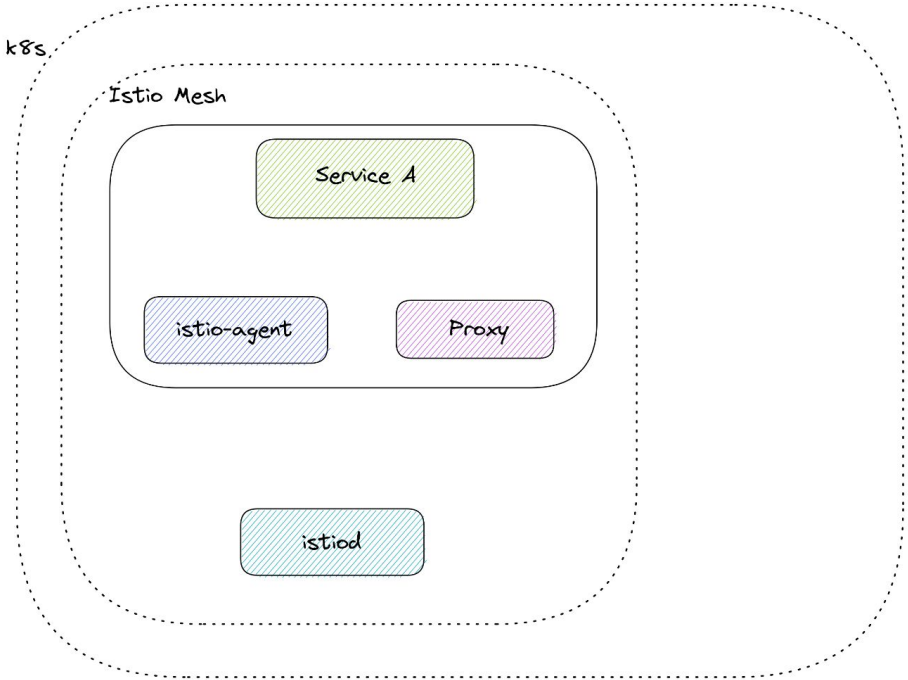
JWT SVID

Service Identity

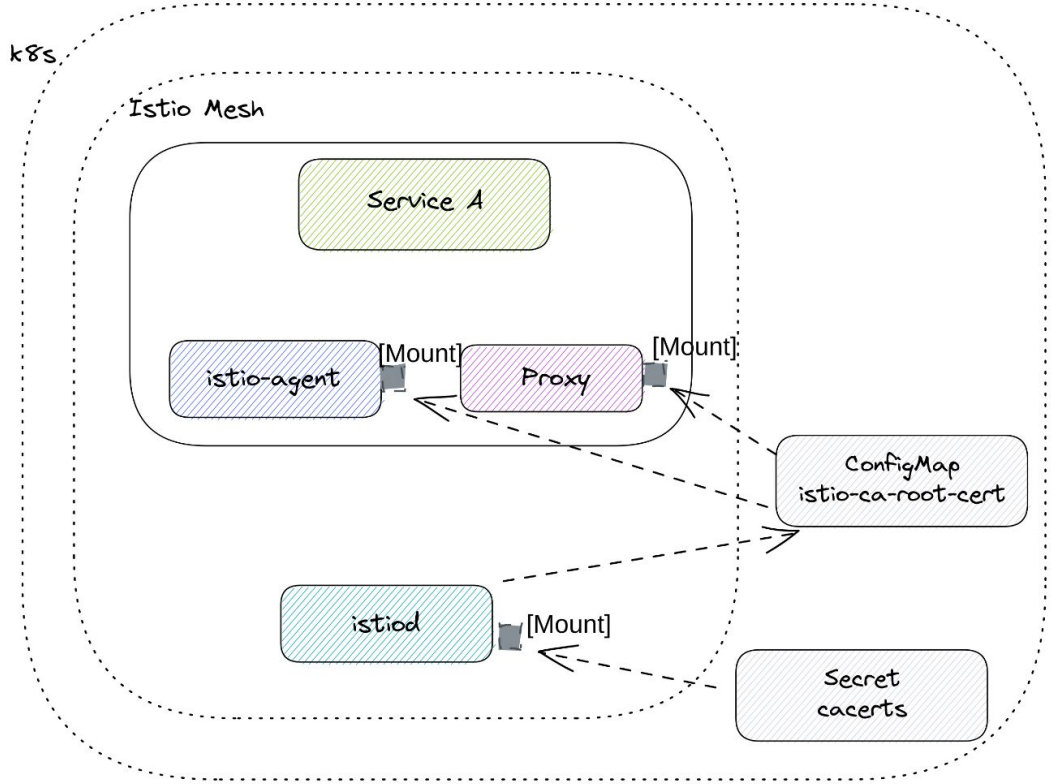
- X.509 certificates using SPIFFE
- Enables Identity and mTLS

Trust Distribution

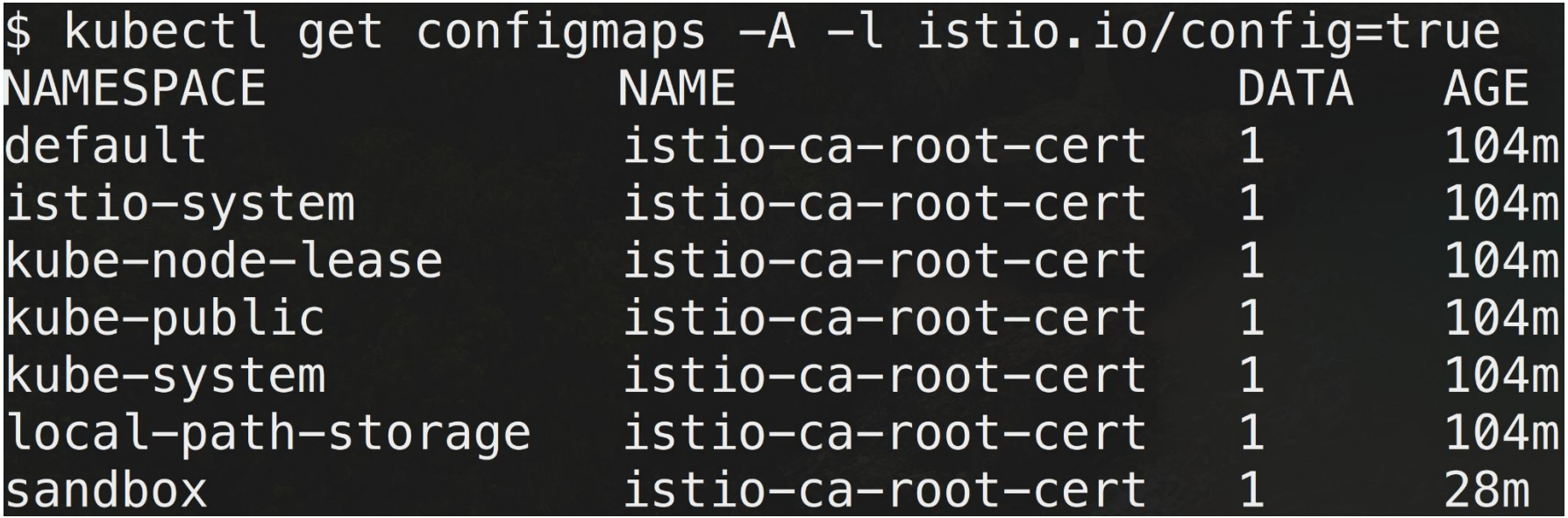
Istio / External CA: Trust Distribution



Istio / External CA: Trust Distribution

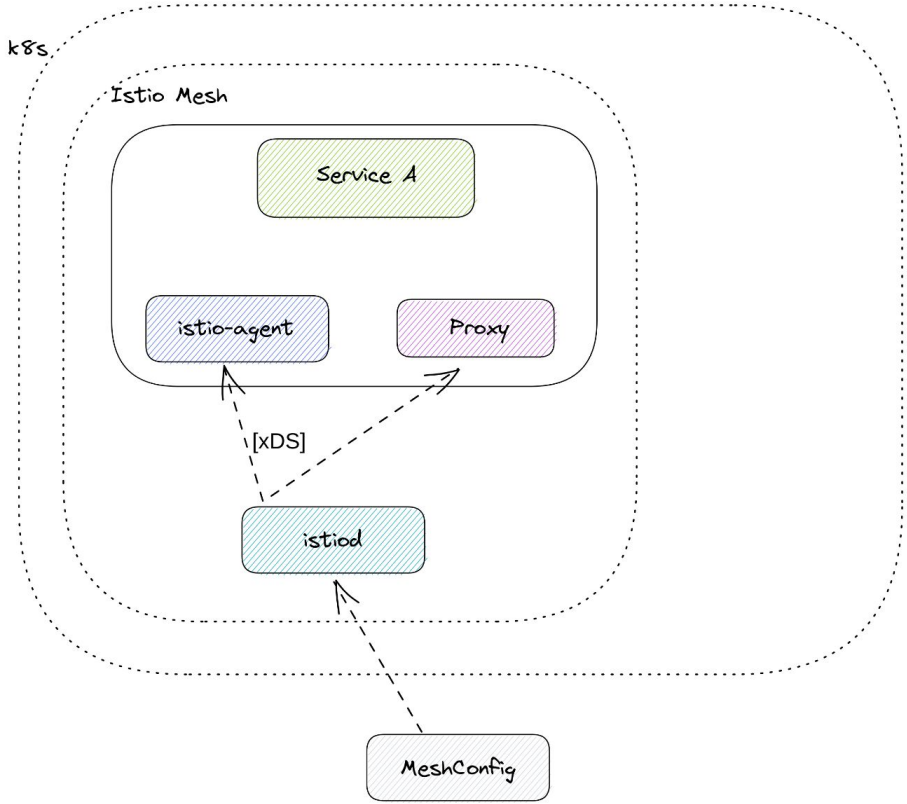


Istio / External CA: Trust Distribution



```
$ kubectl get configmaps -A -l istio.io/config=true
NAMESPACE           NAME                  DATA   AGE
default             istio-ca-root-cert    1      104m
istio-system        istio-ca-root-cert    1      104m
kube-node-lease     istio-ca-root-cert    1      104m
kube-public         istio-ca-root-cert    1      104m
kube-system         istio-ca-root-cert    1      104m
local-path-storage  istio-ca-root-cert    1      104m
sandbox             istio-ca-root-cert    1      28m
```

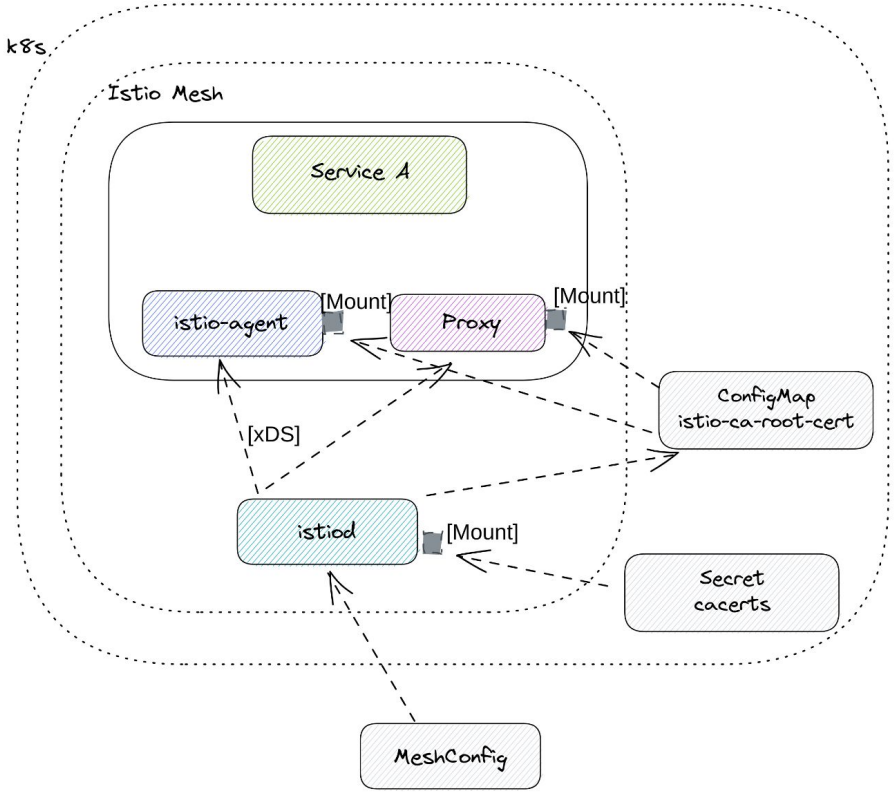
Istio / External CA: Trust Distribution



- Global Mesh Config Trust Anchors
- Pushed out of all workloads by istiod
- Compatible with spiffe compliant servers such as spire

```
apiVersion: install.istio.io/v1alpha1
kind: IstioOperator
metadata:
  namespace: istio-system
spec:
  profile: "demo"
  hub: gcr.io/istio-release
  meshConfig:
    trustDomain: foo.bar
    caCertificates:
    - spiffeBundleUrl: "https://spire.spiffe"
    - pem: |
        -----BEGIN CERTIFICATE-----
        MIIDTDCCAjSgAwIBAgIQUDRiWJ9L2NGL3Ibxa6dk/TANBgkqhkiG9w0BAQsFADBA
        MSswEwYDVQQKEwxjZXJ0LW1hbmFnZXIwFAYDVQQKEw1jbHVzdGVyLmxvY2FsMREw
        DwYDVQQDEwhpc3Rpby1jYTAeFw0yMTA4MDMxMDM2MjBaFw0yMTExMDExMDM2MjBa
        MEAxKzATBgNVBAoTDGNlcnQtbWFuYWdlcjAUBgNVBAoTDWNsdXN0ZXIubG9jYWwx
        ETAPBgNVBAMTCGlzdGlvLWNhMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC
        AQEAuM5xUppoXwoeWWGFhiviwkY5rDxriW5bt9ZO4qYf7/NiBjC68szMxls+XTn3
        SPhx7u0FcBaI7C0a6rYNJxflNuUQ5E+BBjuUd0tnxU6UGIIERPb5ZDtsxZQJq9qh
        eYkRgiDEAi5MKp0+O0uAh9JCFOSq3aZMVS9lQ5ib/S4gNzIiC5skpSVIBrXhtEZV
        foi5VmcTOfS/kDR0avh+yJtxh5vQtUTLf7MpuUBNrhylHF+Q6+M/pD+SpoQwe221
        KHfDLLxAeslHomJBtd2X40lx7JEv9CydjNBxw3eYN1SZ+SMxiJTBT6/KpzSbDOSY
        d7MeH8q3S0kZj09bzhRjlEokBwIDAQABo0IwQDAOBgNVHQ8BAf8EBAMCAqQwDwYD
        VR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUAmTumk1Xkr0eszilLALtnAcGGnMwDQYJ
        KoZIhvcNAQELBQADggEBAC7BS7T9tx337XB0dNsUSL67apumNrfapEzHjxXxPg5F
        dO5JFv7SqJFAh0m90yjaGL3uUKoEb+UCD+7u3J4QMbwCcIL5+i1WmFgg5Zan6FN5
        7W7ImlHqH43Rxw/q6qBrAptQfseLkFQDbRhz6lSqKT7cMUATnik546Fpt2FSle0/
        7+hYZafv+GMeUOageyZAfNVMWBvHH21G1+GNtQWdPf8od3L8QMHTAUoV5xyChR5L
        1GjLfkh4gnhyZI1kR7kVMCTQZ2GR+RLwv9X+Rjai44Sidk7GTeRicwgu9eKKX1oO
        XhihGpVDl38hJCHm2LBwEfrpH6s2Qk1ps4mqlZdmHeU=
        -----END CERTIFICATE-----
```

Istio / External CA: Trust Distribution

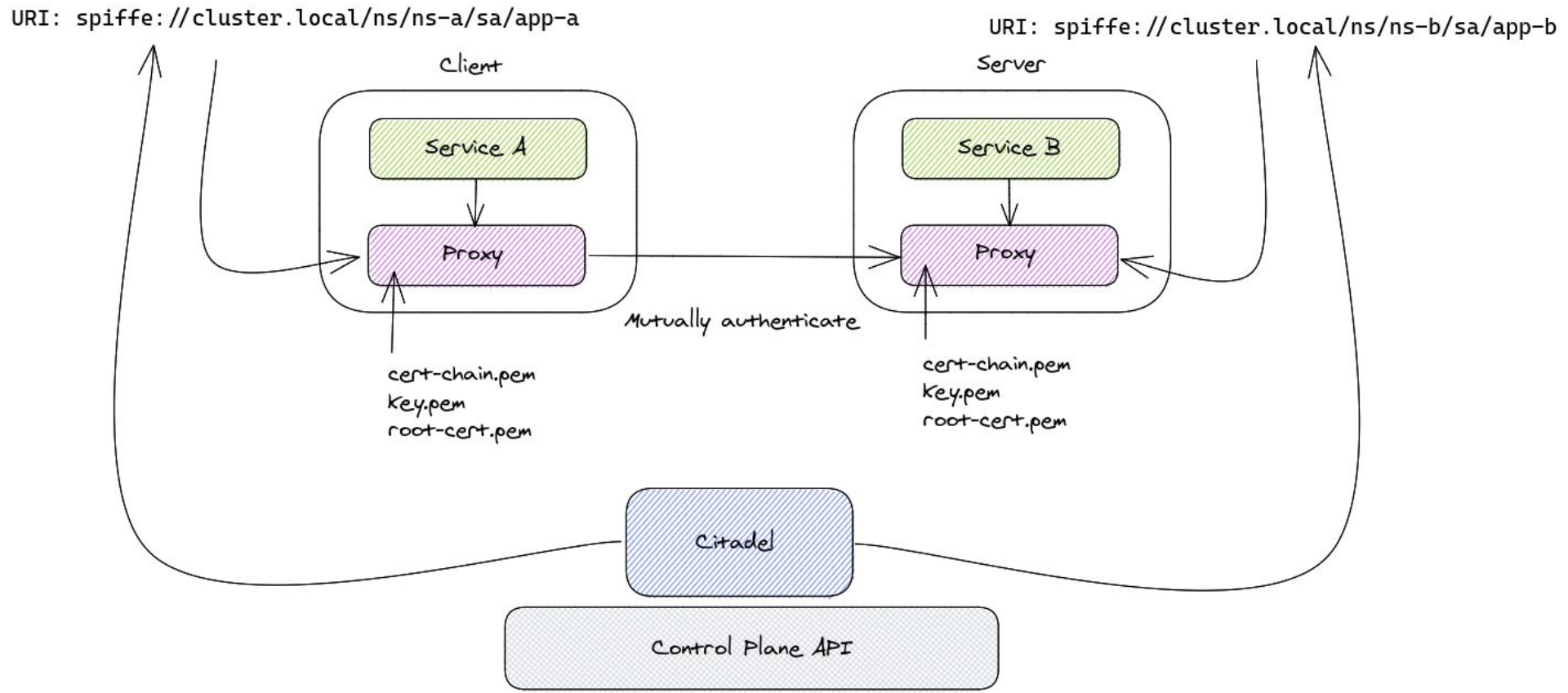


Trust Distribution

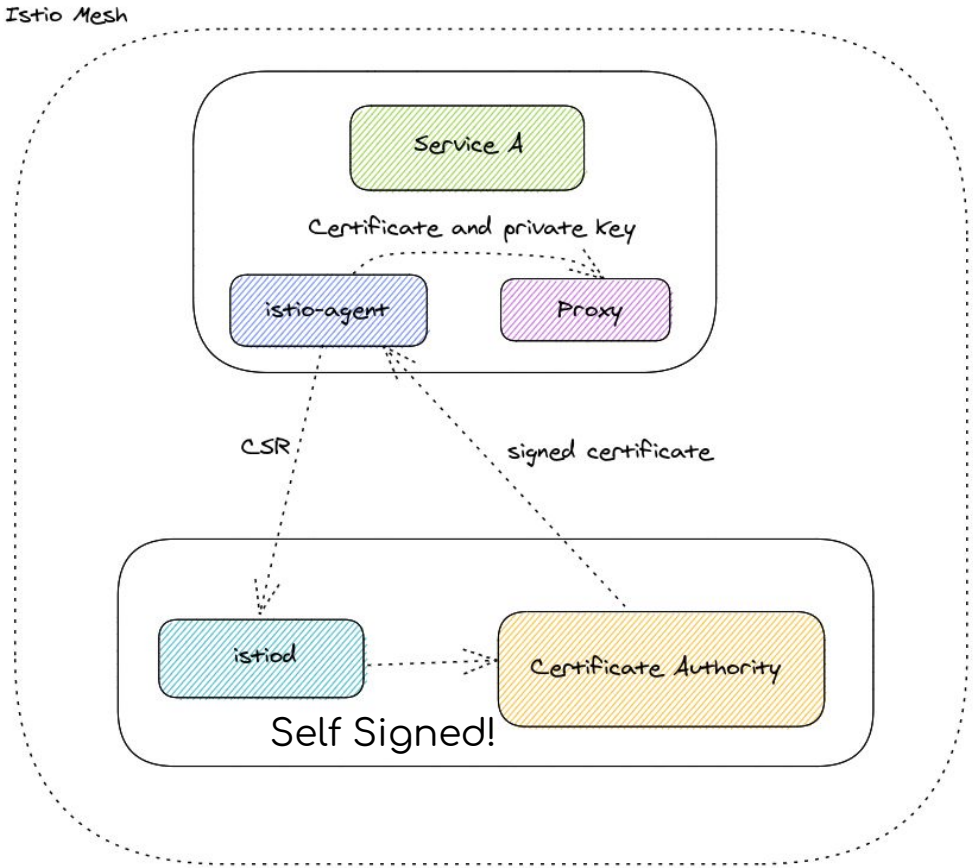
- Trust bundles propagated
- ConfigMap or MeshConfig

Default Installation

Istio / External CA: Default Installation



Istio / External CA: Default Installation



Demo: Default Installation

Default Installation

- CA certificate Self-Signed
- Managed and Signed via Citadel (istiod)

External CAs

Istio / External CA: External CAs

1. Protect the CA private keys (keys to the castle)
2. Observability
3. Policy
4. Integration with existing PKI infrastructure
5. Shared root of trust between meshes/clusters/clouds

Istio / External CA: Plugin in External CA

- Populate cacets Secret in istio-system, ingested by istiod
  - `ca-cert.pem`: the generated intermediate certificates
  - `ca-key.pem`: the generated intermediate key
  - `cert-chain.pem`: the generated certificate chain which is used by istiod
  - `root-cert.pem`: the root certificate

- Behaves the same as default, but with bring your own intermediate

Demo: Plugin External CA

External CAs

- Security / Shared Root
- “Plugin-in” behaves similar to default

Kubernetes CSR

Istio / External CA: Kubernetes CSR

```yaml
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: isolated-issuer-root-csr
  annotations:
    experimental.cert-manager.io/request-duration: 1h0m0s
    experimental.cert-manager.io/request-is-ca: "false"
spec:
  groups:
  - system:masters
  - system:authenticated
  request: ...
  signerName: issuers.isolated-issuer.jetstack.io/istio-system.root-ca-issuer
  usages:
  - signing
  - key encipherment
  username: kubernetes-admin
status:
  certificate: ...
  conditions:
  - lastTransitionTime: "2021-08-02T18:23:52Z"
    lastUpdateTime: "2021-08-02T18:23:52Z"
    message: This CSR was approved by kubectl certificate approve.
    reason: KubectlApprove
    status: "True"
    type: Approved
```

Istio / External CA: Kubernetes CSR

1. Requestor

```yaml
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: isolated-issuer-root-csr
  annotations:
    experimental.cert-manager.io/request-duration: 1h0m0s
    experimental.cert-manager.io/request-is-ca: "false"
spec:
  groups:
  - system:masters
  - system:authenticated
  username: kubernetes-admin
  request: ...
  signerName: issuers.isolated-issuer.jetstack.io/istio-system.root-ca-issuer
  usages:
  - signing
  - key encipherment
status:
  certificate: ...
  conditions:
  - lastTransitionTime: "2021-08-02T18:23:52Z"
    lastUpdateTime: "2021-08-02T18:23:52Z"
    message: This CSR was approved by kubectl certificate approve.
    reason: KubectlApprove
    status: "True"
    type: Approved
```

Istio / External CA: Kubernetes CSR

1. Requestor

2. Signer

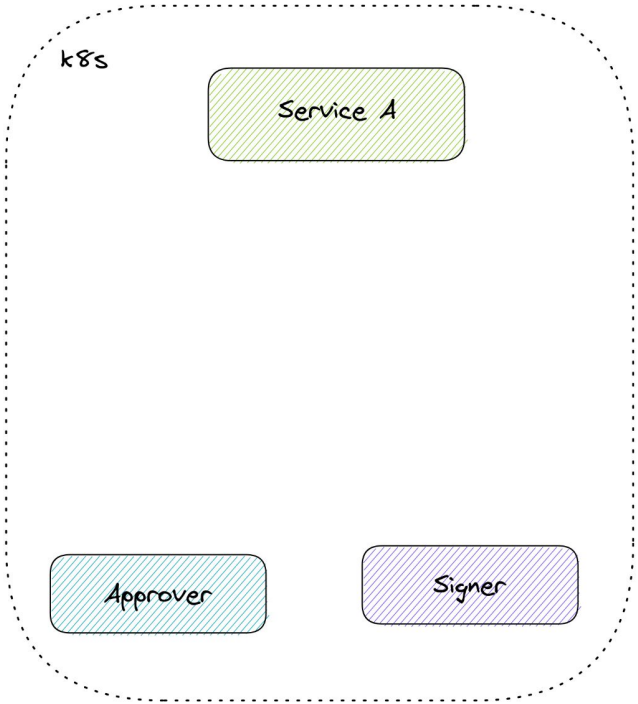
```yaml
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: isolated-issuer-root-csr
  annotations:
    experimental.cert-manager.io/request-duration: 1h0m0s
    experimental.cert-manager.io/request-is-ca: "false"
spec:
  groups:
  - system:masters
  - system:authenticated
  username: kubernetes-admin
  request: ...
  signerName: issuers.isolated-issuer.jetstack.io/istio-system.root-ca-issuer
  usages:
  - signing
  - key encipherment
status:
  certificate: ...
  conditions:
  - lastTransitionTime: "2021-08-02T18:23:52Z"
    lastUpdateTime: "2021-08-02T18:23:52Z"
    message: This CSR was approved by kubectl certificate approve.
    reason: KubectlApprove
    status: "True"
    type: Approved
```

Istio / External CA: Kubernetes CSR

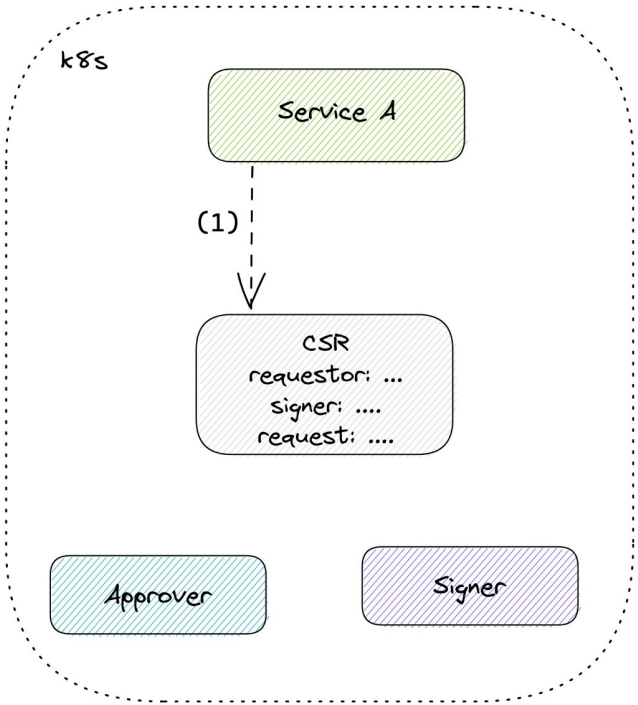
1. Requestor
2. Signer
3. Approver

```yaml
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: isolated-issuer-root-csr
  annotations:
    experimental.cert-manager.io/request-duration: 1h0m0s
    experimental.cert-manager.io/request-is-ca: "false"
spec:
  groups:
  - system:masters
  - system:authenticated
  username: kubernetes-admin
  request: ...
  signerName: issuers.isolated-issuer.jetstack.io/istio-system.root-ca-issuer
  usages:
  - signing
  - key encipherment
status:
  certificate: ...
  conditions:
  - lastTransitionTime: "2021-08-02T18:23:52Z"
    lastUpdateTime: "2021-08-02T18:23:52Z"
    message: This CSR was approved by kubectl certificate approve.
    reason: KubectlApprove
    status: "True"
    type: Approved
```

Istio / External CA: Kubernetes CSR

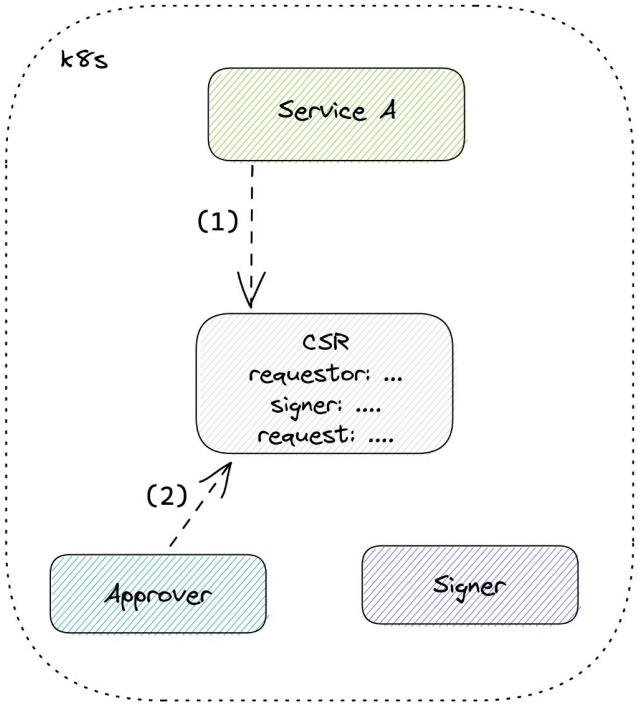


Istio / External CA: Kubernetes CSR



1. CSR requested by Service A (requestor)

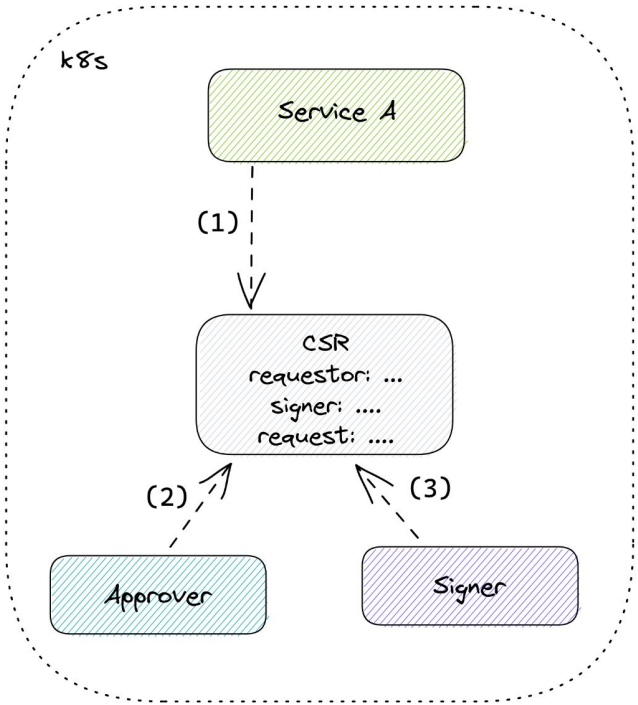
Istio / External CA: Kubernetes CSR



1. CSR requested by Service A (requestor)

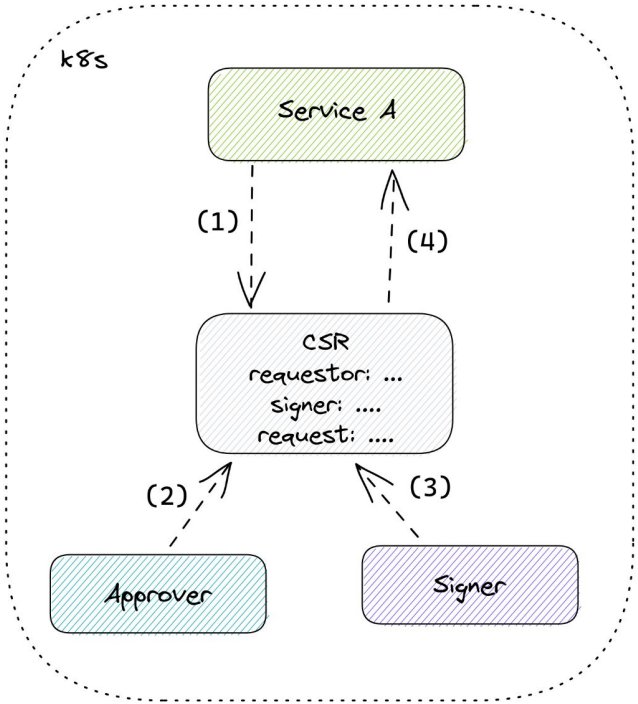
2. Approver approves the request based on contents and requestor

Istio / External CA: Kubernetes CSR



1. CSR requested by Service A (requestor)
2. Approver approves the request based on contents and requestor
3. Signer Signes the Approved request

Istio / External CA: Kubernetes CSR



1. CSR requested by Service A (requestor)
2. Approver approves the request based on contents and requestor
3. Signer Signs the Approved request
4. Signed certificate read by Service A

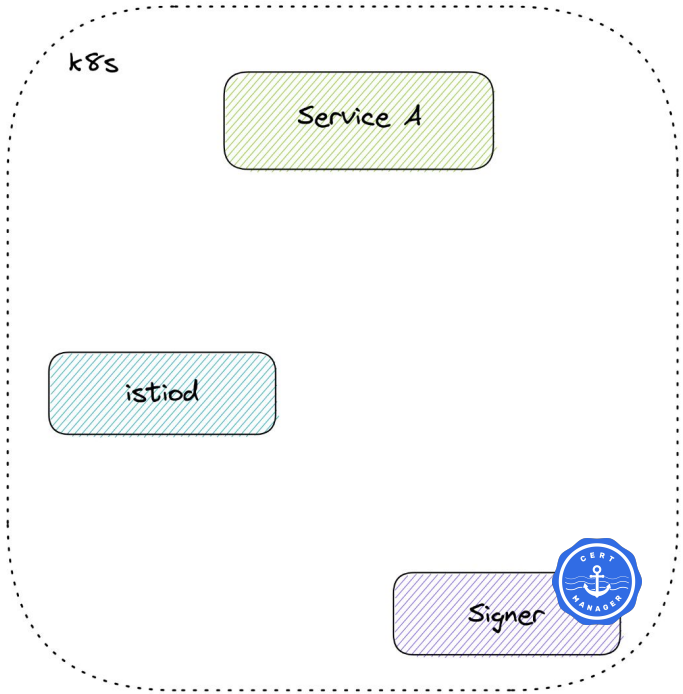
Istio / External CA: Kubernetes CSR

- cert-manager https://github/jetstack/cert-manager
- De-facto certificates controller for Kubernetes
- Issuer or ClusterIssuer describe a CA or “Signer”
- Support for signing Kubernetes CSRs

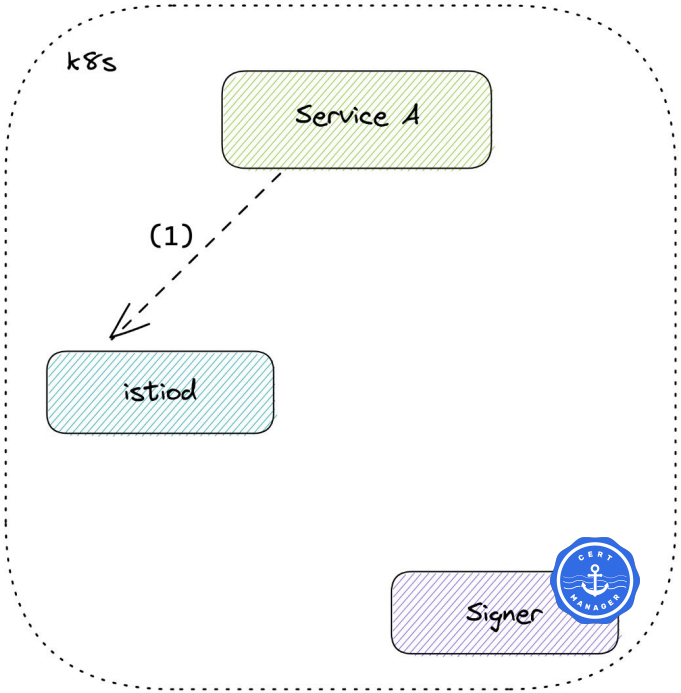
Istio / External CA: Kubernetes CSR


```yaml
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
  name: vault
  namespace: istio-system
spec:
  vault:
    auth:
      kubernetes:
        role: istio-ca
        mountPath: /v1/auth/kubernetes
        secretRef:
          name: vault-issuer-token-4grct
          key: token
    path: pki/sign/istio-ca
    server: http://vault.vault:8200
```

Istio / External CA: Kubernetes CSR

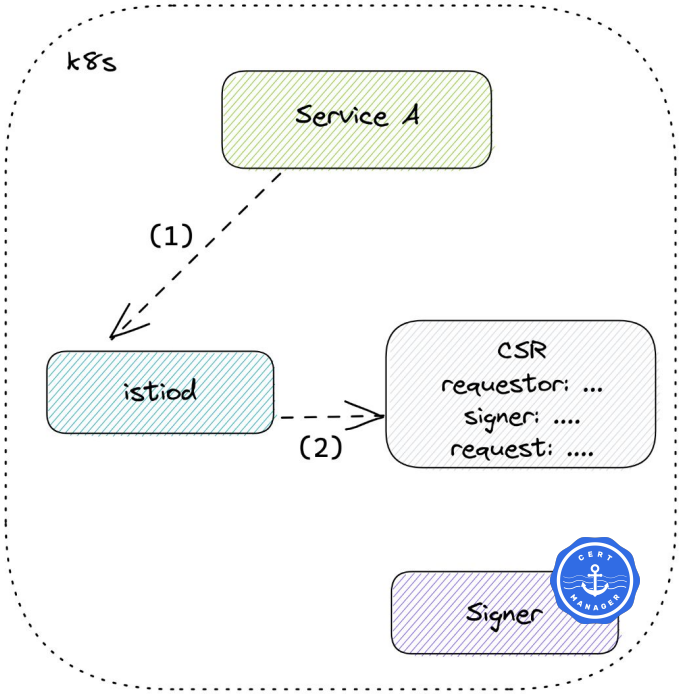


Istio / External CA: Kubernetes CSR



1. CSR requested by Service A to istiod using gRPC

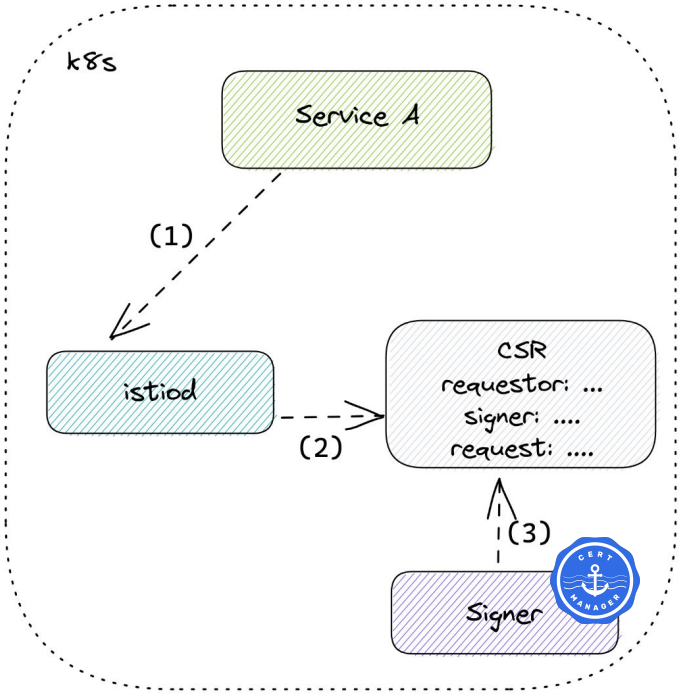
Istio / External CA: Kubernetes CSR



1. CSR requested by Service A to istiod using gRPC

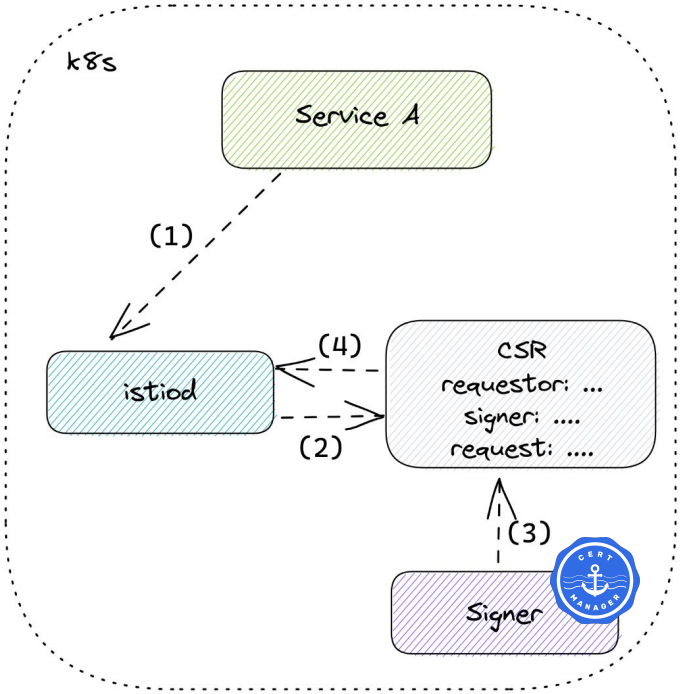
2. istiod creates CSR (requestor), Approves the request

Istio / External CA: Kubernetes CSR



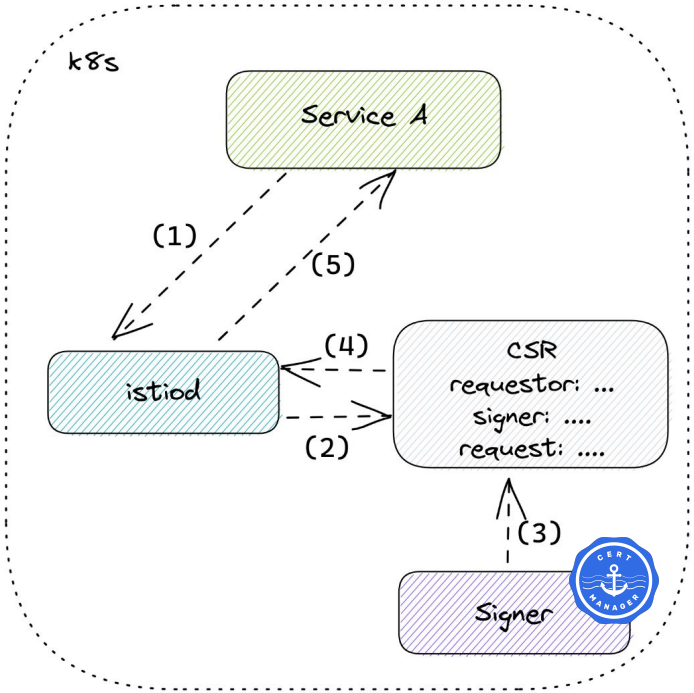
1. CSR requested by Service A to istiod using gRPC
2. istiod creates CSR (requestor), Approves the request (approver)
3. cert-manager signs request

Istio / External CA: Kubernetes CSR



1. CSR requested by Service A to istiod using gRPC
2. istiod creates CSR (requestor), Approves the request
3. cert-manager signs request
4. istiod reads signed certificate

Istio / External CA: Kubernetes CSR



1. CSR requested by Service A to istiod using gRPC
2. istiod creates CSR (requestor), Approves the request
3. cert-manager signs request
4. istiod reads signed certificate
5. istiod responds to service with signed certificate

Demo: Kubernetes CSR

Kubernetes CSR

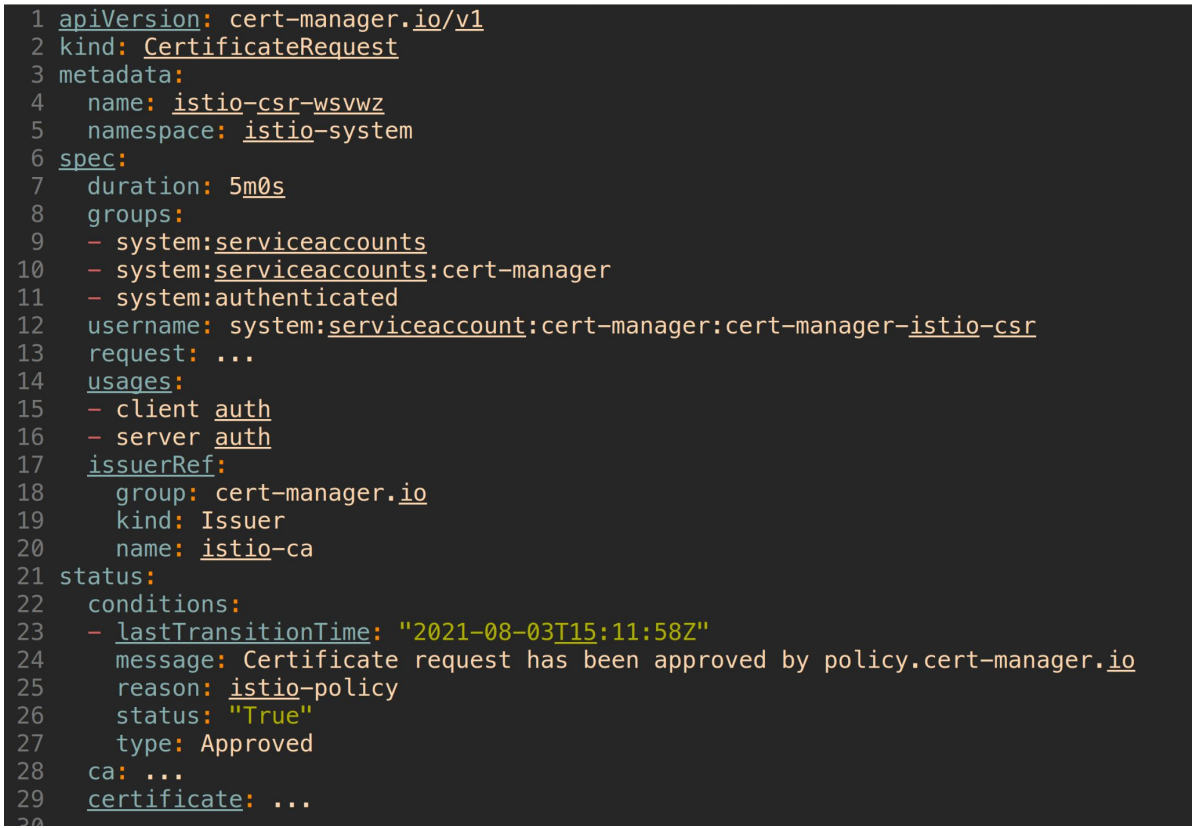
- Leverage Kubernetes resource for Signing
- istiod acts as RA

cert-manager / istio-csr

Istio / External CA: cert-manager / istio-csr

- istio-csr https://github/cert-manager/istio-csr
- Acts as the RA for Istio certificates
- Takes care of trust distribution
- cert-manager acts as CA
- Distributes trust for Isito
- Compatible with any Issuer that can sign SPIFFE certificates

Istio / External CA: cert-manager / istio-csr



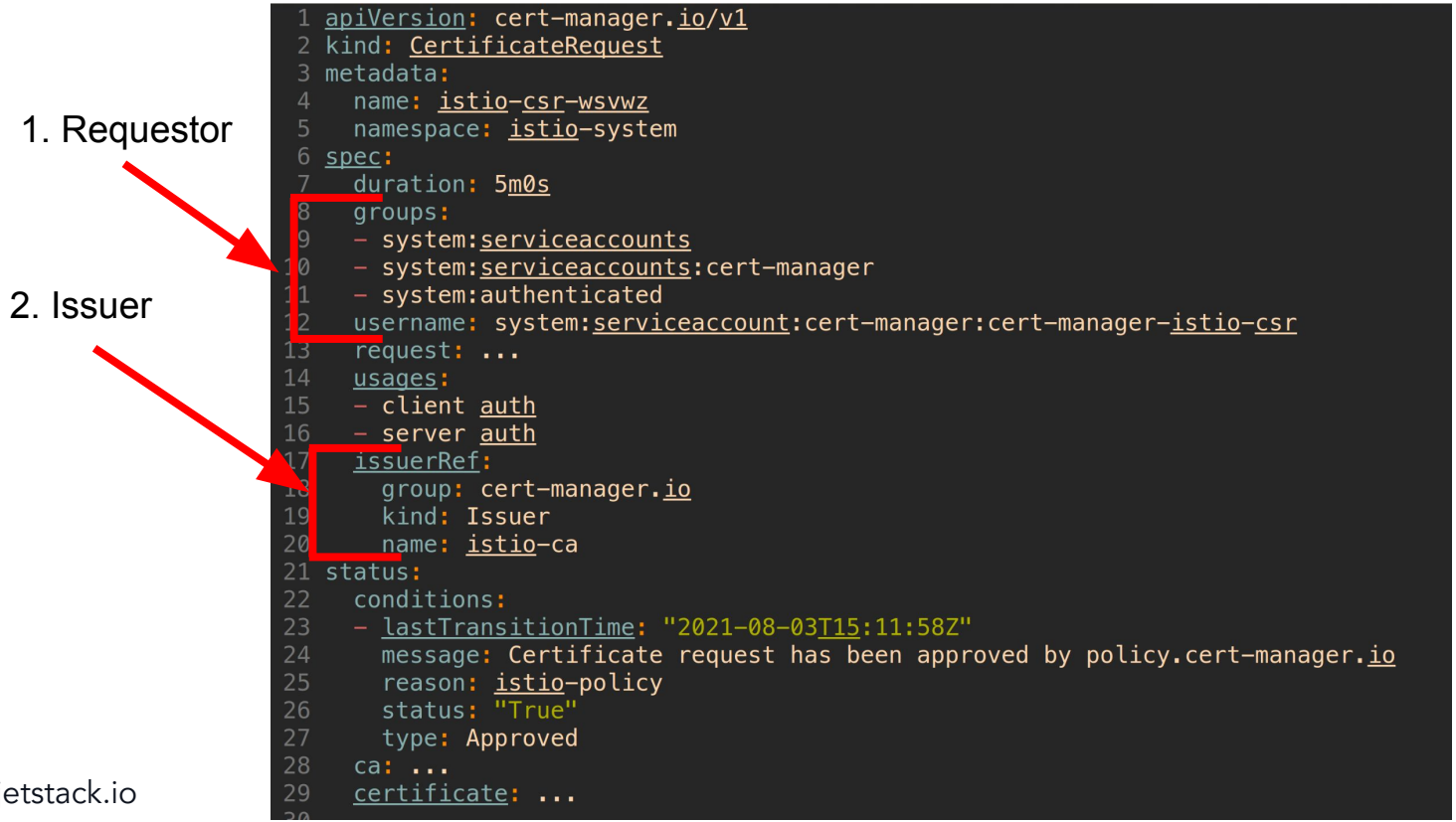
```yaml
apiVersion: cert-manager.io/v1
kind: CertificateRequest
metadata:
  name: istio-csr-wsvwz
  namespace: istio-system
spec:
  duration: 5m0s
  groups:
  - system:serviceaccounts
  - system:serviceaccounts:cert-manager
  - system:authenticated
  username: system:serviceaccount:cert-manager:cert-manager-istio-csr
  request: ...
  usages:
  - client auth
  - server auth
  issuerRef:
    group: cert-manager.io
    kind: Issuer
    name: istio-ca
status:
  conditions:
  - lastTransitionTime: "2021-08-03T15:11:58Z"
    message: Certificate request has been approved by policy.cert-manager.io
    reason: istio-policy
    status: "True"
    type: Approved
  ca: ...
  certificate: ...
```

Istio / External CA: cert-manager / istio-csr

1. Requestor

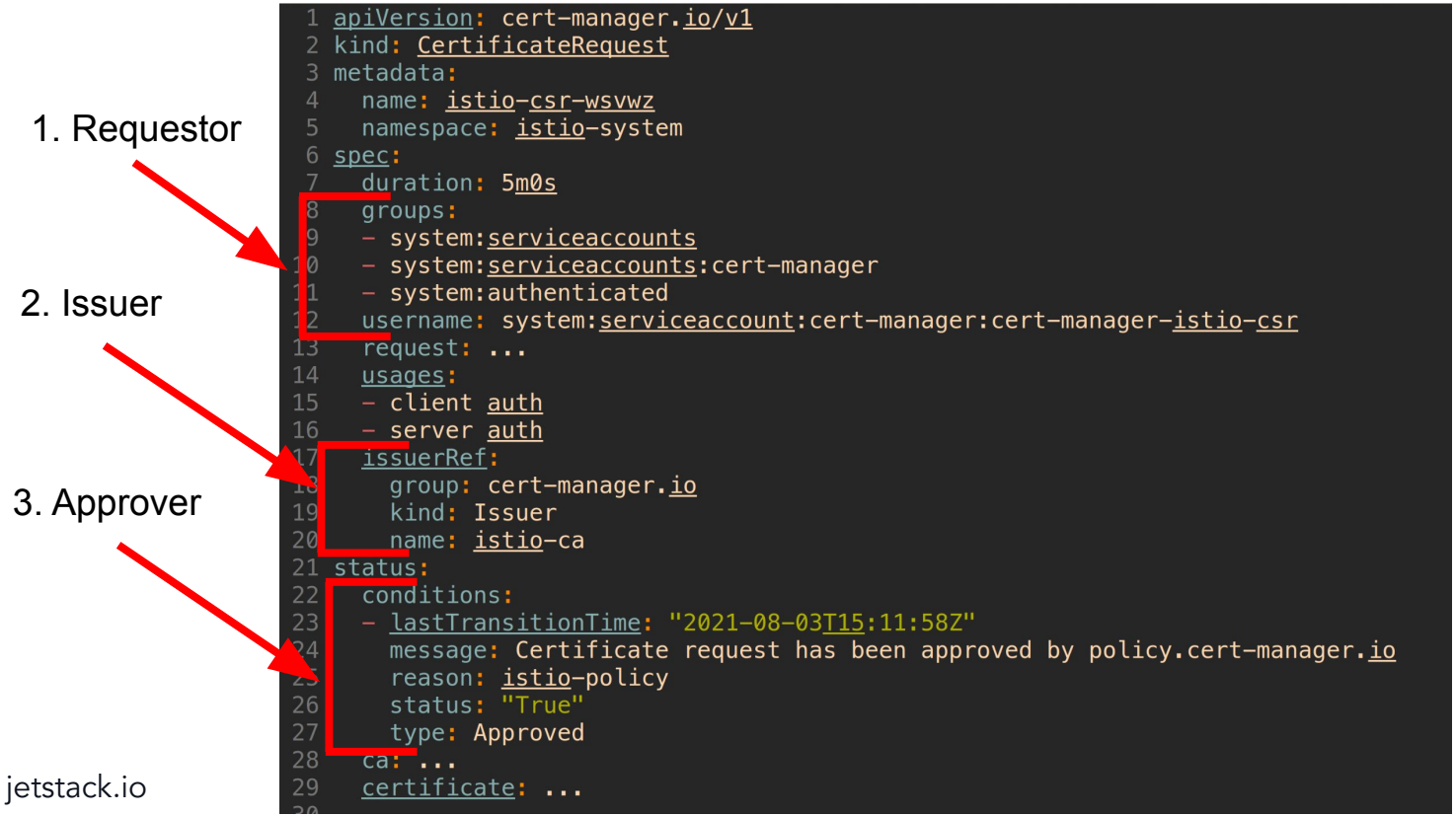
```yaml
apiVersion: cert-manager.io/v1
kind: CertificateRequest
metadata:
  name: istio-csr-wsvwz
  namespace: istio-system
spec:
  duration: 5m0s
  groups:
  - system:serviceaccounts
  - system:serviceaccounts:cert-manager
  - system:authenticated
  username: system:serviceaccount:cert-manager:cert-manager-istio-csr
  request: ...
  usages:
  - client auth
  - server auth
  issuerRef:
    group: cert-manager.io
    kind: Issuer
    name: istio-ca
status:
  conditions:
  - lastTransitionTime: "2021-08-03T15:11:58Z"
    message: Certificate request has been approved by policy.cert-manager.io
    reason: istio-policy
    status: "True"
    type: Approved
  ca: ...
  certificate: ...
```

Istio / External CA: cert-manager / istio-csr

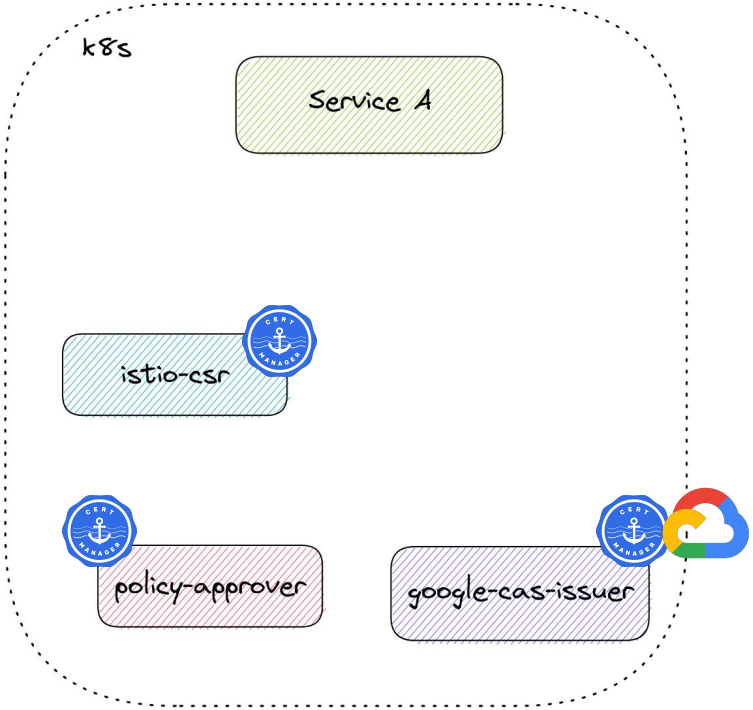



```yaml
apiVersion: cert-manager.io/v1
kind: CertificateRequest
metadata:
  name: istio-csr-wsvwz
  namespace: istio-system
spec:
  duration: 5m0s
  groups:
  - system:serviceaccounts
  - system:serviceaccounts:cert-manager
  - system:authenticated
  username: system:serviceaccount:cert-manager:cert-manager-istio-csr
  request: ...
  usages:
  - client auth
  - server auth
  issuerRef:
    group: cert-manager.io
    kind: Issuer
    name: istio-ca
status:
  conditions:
  - lastTransitionTime: "2021-08-03T15:11:58Z"
    message: Certificate request has been approved by policy.cert-manager.io
    reason: istio-policy
    status: "True"
    type: Approved
  ca: ...
  certificate: ...
```

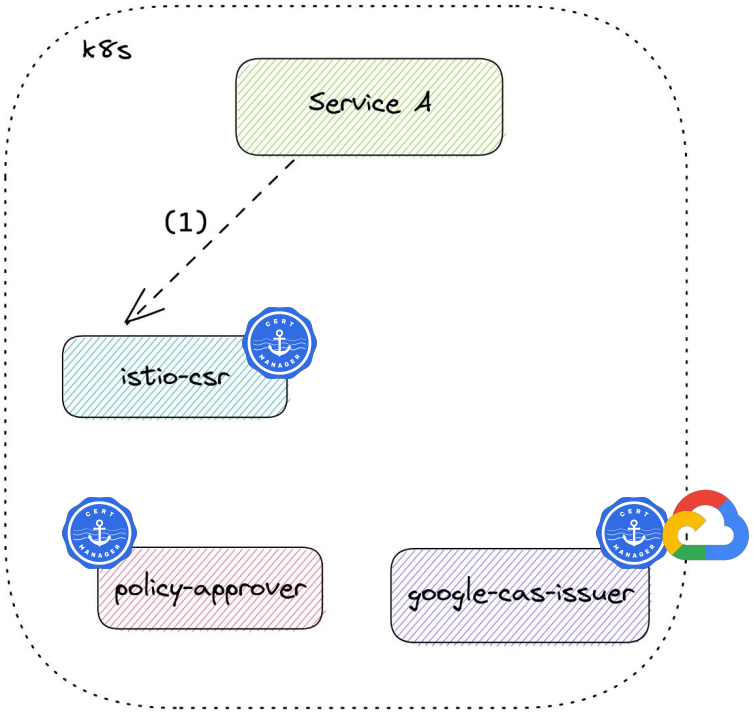
Istio / External CA: cert-manager / istio-csr



Istio / External CA: cert-manager / istio-csr

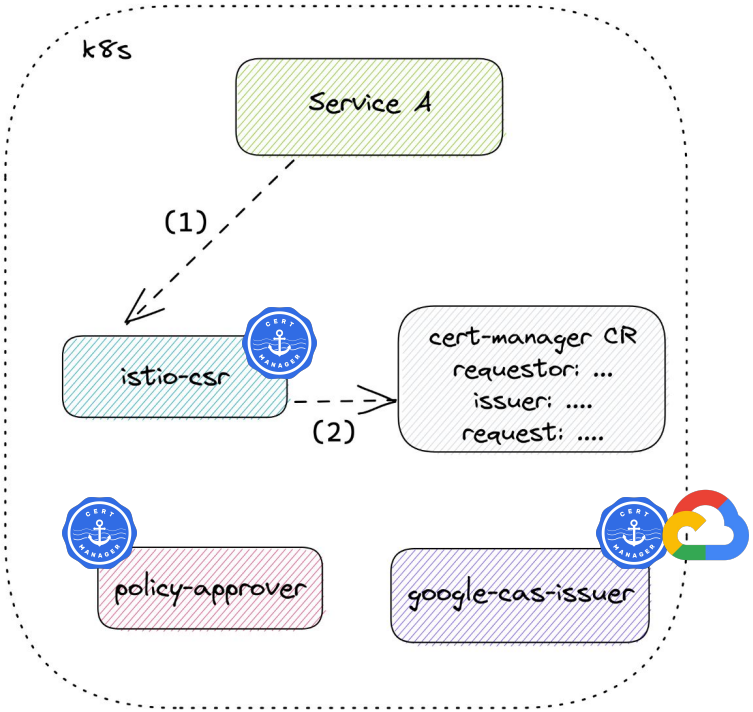


Istio / External CA: cert-manager / istio-csr



1. CSR requested by Service A to istio-csr using gRPC

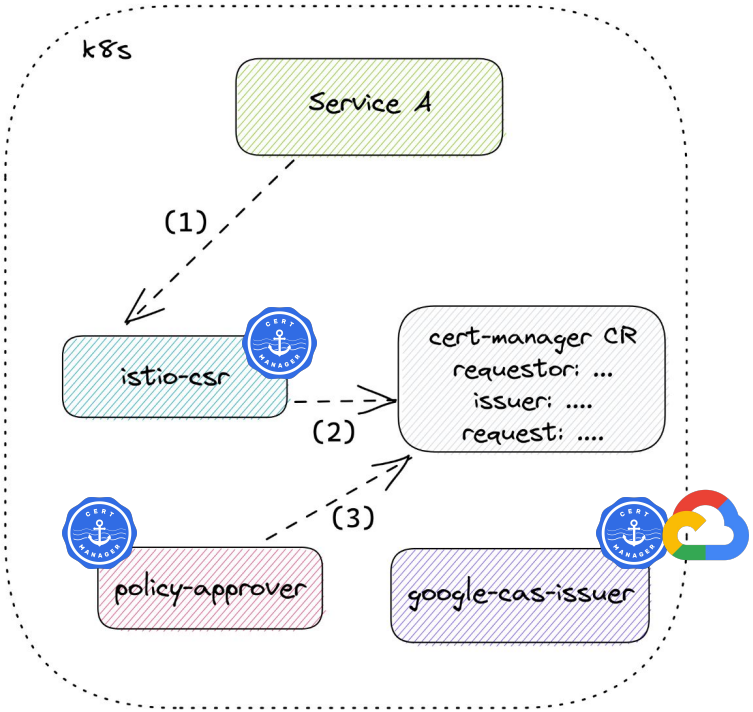
Istio / External CA: cert-manager / istio-csr



1. CSR requested by Service A to istio-csr using gRPC

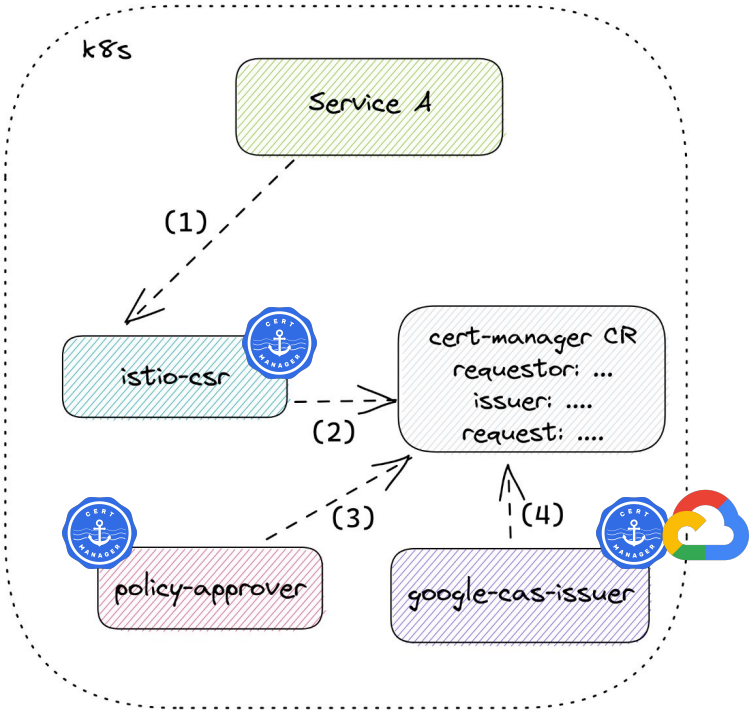
2. istio-csr (requestor) creates cert-manager CertificateRequest

Istio / External CA: cert-manager / istio-csr



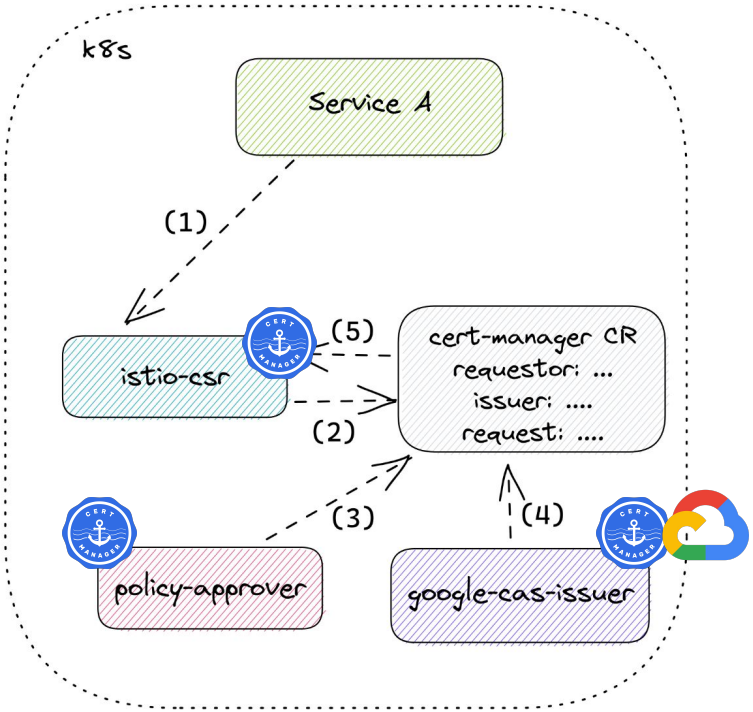
1. CSR requested by Service A to istio-csr using gRPC
2. istio-csr (requestor) creates cert-manager CertificateRequest
3. policy-approver Approves the request

Istio / External CA: cert-manager / istio-csr



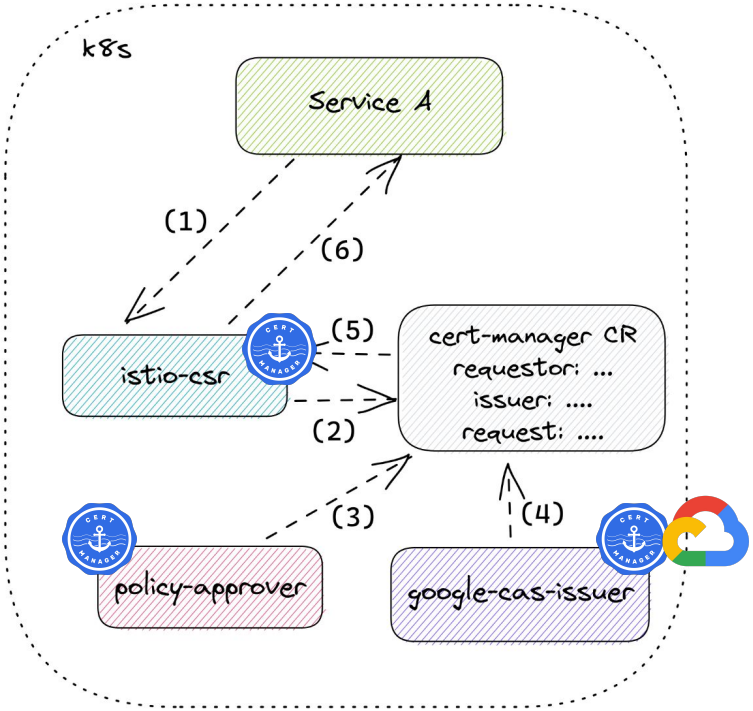
1. CSR requested by Service A to istio-csr using gRPC
2. istio-csr (requestor) creates cert-manager CertificateRequest
3. policy-approver Approves the request
4. google-cas-issuer signs the request

Istio / External CA: cert-manager / istio-csr



1. CSR requested by Service A to istio-csr using gRPC
2. istio-csr (requestor) creates cert-manager CertificateRequest
3. policy-approver Approves the request
4. google-cas-issuer signs the request
5. Istio-csr reads the signed request

Istio / External CA: cert-manager / istio-csr



1. CSR requested by Service A to istio-csr using gRPC
2. istio-csr (requestor) creates cert-manager CertificateRequest
3. policy-approver Approves the request
4. google-cas-issuer signs the request
5. Istio-csr reads the signed request
6. istio-csr responds to service with signed certificate

Demo: cert-manager / istio-csr

cert-manager / istio-csr

- Leverage cert-manager for signing
- Use existing cert-manager Issuers and tooling

Istio / External CA: Fin

- Istio uses SPIFFE documents for identity
- Istio has different methods for trust distribution
- Default installation uses self-signed certificates
- External CAs bring Security and shared roots

Istio / External CA: Fin

- Istio uses SPIFFE documents for identity
- Istio has different methods for trust distribution
- Default installation uses self-signed certificates
- External CAs bring Security and shared roots
- External CAs can be “plugged in” via Secrets
- Istio can use Kubernetes CSRs for signing
- cert-manager/istio-csr brings integration of cert-manager with Istio

Istio / External CA: Fin

- Onward… Istio Workload Certificate API


JETSTACK
Thank you.
joshua.vanleeuwen @jetstack.io
@JoshVanL
jetstack.io